AN INTRODUCTION TO PROTOCOLS FOR REAL-TIME COMMUNICATIONS IN PACKET SWITCHED NETWORKS

DECEMBER 1997

AUTHOR

TOMI YLETYINEN *

INSTRUCTOR

M.SC. MARKO LUOMA *

* HELSINKI UNIVERSITY OF TECHNOLOGY

LABORATORY OF TELECOMMUNICATIONS TECHNOLOGY

Abstract

The packet networks are used today for an application they are were not originally designed for: networked real-time multimedia. A lot of research effort has gone into making this possible: new real-time protocol concepts are evolving. This study gives an introduction to protocols designed with real-time needs in mind

.

An Introduction to Protocols for Real-Time Communications in Packet Switched Networks

Tomi Yletyinen

Laboratory of Telecommunications Technology

Faculty of Electrical and Communication Engineering

Helsinki University of Technology

Contents

List of Figures

List of Tables

List of Abbreviations

ABR	Available Bit Rate
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol
BGP	Border Gateway Protocol
CBR	Constant Bit Rate
FDDI	Fiber Distributed Data Interface
FF	Fixed-Filter, style of reservations in RSVP
FlowSpec	Flow Specification
HS	Hard State
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Multicast Protocol
IGP	Interior Gateway Protocol
IIS	Internet Integrated Services
IP	Internet Protocol
IPANA	IP and Atm advanced Network Architectures
IS	Integrated Services
ISA	Integrated Services Architecture
ISO	International Standards Organization
ITU	International Telecommunications Union
LAN	Local Area Network
LANE	LAN Emulation
LIS	Logical IP Subnetworks
MARS	Multicast Address Resolution Server

MIT	Massachusetts Institute of Technology
MOS	Mean Opinion Score
MPOA	Multi Protocol Over Atm
MSTP	Multimedia Synchronization Transport Protocol.
NAP	Network Access Point
nrtVBR	Non-real-time Variable Bit Rate
OCPN	Object Composition Petri Net model
PARC	Xerox Palo Alto Research Center
PDU	Protocol Data Unit
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RARP	Reverse ARP
RCAP	connection management protocol in Tenet Suite
RFC	Request For Comments
RMTP	Real-time Message Transport Protocol (Tenet Suite)
Rspec	A flow specification parameter specifying the QoS desired
RSVP	resource ReSerVation Protocol
RTCMP	Real-Time Control Message Protocol (Tenet Suite)
RTCP	RTP control protocol
RTIP	Real-Time Internetwork Protocol (Tenet Suite)
RTP	Real-time Transport Protocol
RTSM	Real-time Synchronization Model
rtVBR	Real-time Variable Bit Rate
SCMP	Stream Control Message Protocol (ST)
SE	Shared-Explicit, style of reservations in RSVP

SS	Soft State
ST	internet STream protocol
ST	data transport protocol in ST
ST-II+	internet STream protocol version two
SVC	Switched Virtual Circuit
TCP	Transmission Control Protocol
Tspec	A parameter specifying the flow type
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
UNI	User to Network Interface
USC/ISI	University of Southern California Information Sciences Institute
VC	Virtual Cicuit
WF	Wildcard-Filter , syle of reservations in RSVP
WFQ	Weighted Fair Queuing
VP	Virtual Path

Introduction

The advances in technology during the last years have made networked multimedia a reality. A wealth of applications such as audio and video conferencing, shared workspace and group ware systems are fast making their way to the desktop at an affordable price. They are used over packet switched networks, company LANs, intranets, extranets and even over the Internet.

Two major problems of packet switched and cell switched networks supporting multimedia applications are random delay and lost packets due to buffer overflow. The temporal relationships among and in media maybe destroyed at the receiver even if they were sent in synchronization. Since each medium is usually transmitted in a separate connection the quality of service of the connections may be different resulting in different delay characteristics. The random delay of the network causes the packets sent at constant intervals between packets to arrive at the receiver with variable spacing between packets, i.e. the transmission delay of the packets seen by the receiver varies. Some packets never arrive: the network looses packets more or less randomly. When the network load is high the amount of packets lost rises. Packet losses as high as 1-10% are not uncommon in some areas of the Internet of today.

The purpose of this study is to serve as an introduction to the various protocols for real-time communications in packet networks. The protocols presented concern the IP world. Native ATM solutions provide possibly the best delay and delay variance control, but they are outside the scope of this study. Solutions such as RSVP and ST-II+ that use a QoS guaranteeing link layer (such as ATM) are handled, but the specific technologies that concern the mappings are not presented. Examples of such technologies are IP-switching, Multiprotocol over ATM (MPOA) and Label Switching.

Some of the current real-time communication protocols are presented in chapters one and two. In chapter one protocols designed for the purpose of session level synchronization such as RTP, the Real-time Transport Protocol and MSTP, the Multimedia Synchronization Transport Protocol, are presented.

After the session layer protocols follows an introduction to the protocols that try to provide better quality of service for real-time applications through reservation of resources in the network. One of the first attempts ever made at addressing the real-time requirements in packet switched networks, namely Internet Stream Protocol (ST). A similar experimental protocol was developed by a group of scientists in University of California, Berkeley Computer Science Division called the Tenet Suite. The Tenet Suite goes even further than ST along the path of providing performance guarantees. A group of researchers in MIT, Xerox PARC, USC/ISI and Stanford have designed a five module approach to the problem, which goes now by the name Internet Integrated Services Architecture (IIS or ISA) and is being developed in Internet Engineering Task Force. IIS is presented in chapter 2.3.

1. Session Layer Protocols

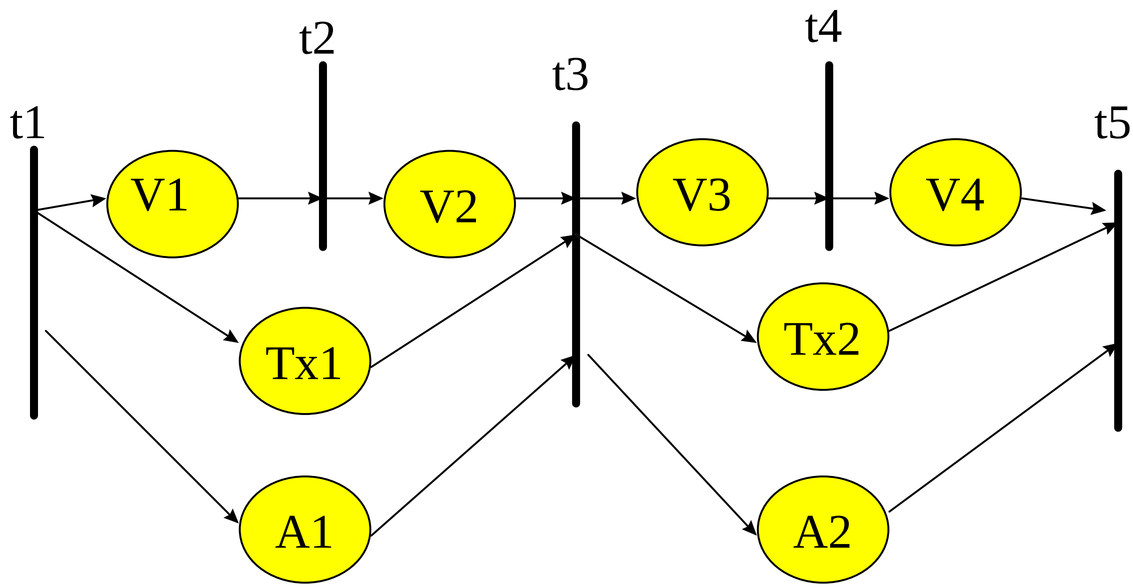
The Internet has been based on the philosophy that most traffic is elastic in the sense that it does not require exact delivery times. The traditional Internet traffic school believes that this holds still reasonably well most of the time, and that by making real-time applications more adaptive to changing conditions in the network the quality of communication can be acceptable. This has been proven right to some extent by the design of new protocols that aim to give applications means to synchronize media and to adapt by following the changes in the level of congestion in the network. When the application knows the approximate delay, an estimate of the delay variance and packet loss rate it can for example adjust the play-out delay, the delay added at the receiver to smooth out packet arrival delay variance, close to the optimum.

This chapter starts by a short introduction to how synchronization can be modeled with Petri nets, and how the traditional model be improved with a slightly modified version of Petri nets. This will be followed by an introduction of an experimental protocol, the Multimedia Synchronization Transport Protocol that aims at optimal synchronization between different media at the receiver.

The MSTP will be followed by a presentation of the now "de facto" standard RTP. RTP is also used in ITU-T's video conferencing recommendation H.323. An integral part of H.323 is the recommendation H.255.0 which covers protocols and message formats. The most important parts of H.225.0 in regard to this study are presented in 1.4.

1.1. Synchronization models with Petri nets

Much work has been done in describing the synchronization model for multimedia applications. All these models try specify the temporal relationships between streams of data with different levels of abstraction. Among the most common are the Petri-net based models. One such model is the object composition Petri net model (OCPN). It is good for describing the arbitrary temporal relationships among media. The OCPN can be used both for stored-data and live applications. The extended OCPN can also specify communications functions. However, the Petri net models are not sufficient for packet networks, with variable amounts of delay. In multimedia applications this has to be taken into account, and therefore the model used should be able to deal with late transmission of packets. A late transmission packet is a packet that fails to arrive at the destination in time [1].



V:video, Tx:text, A:Audio

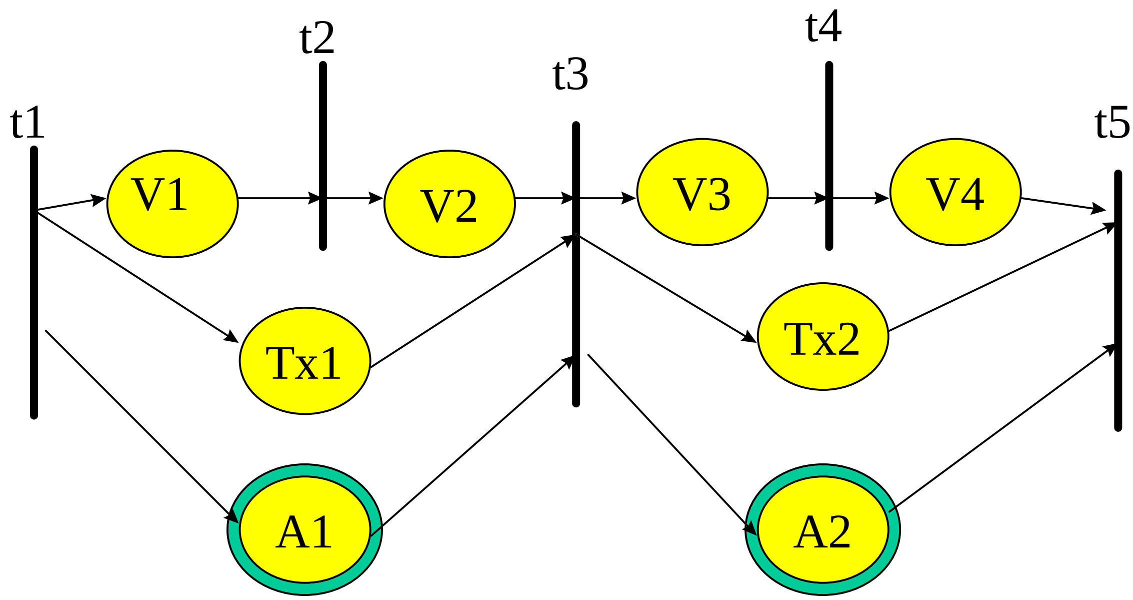
Figure 1: A Petri net based synchronization model.

Figure one shows an example application abstraction. In transition from state t1 to t3 the application shows two video clips V1 and V2 while continuously playing audio A1 and showing an overlay text (subtitle) Tx1.Transition from state t3 on will take place after V2, Tx1 and A1 finish playing. Audio is very delay variance sensitive. If we hold the playing of A2 very too long after A1 there will be noticeable clicks in the audio. The dropping or holding of video frames or random delays between video frames on the other hand will only be barely noticeable to the human eye, and therefore will be tolerable. In the event of late transmission of V2 , A2 can not be played earlier in this model, even if it had arrived. The proper action would be to discard V2 and start playing A2. The Petri Net is therefore only sufficient in perfect network conditions.

There are at least to possible places for implementation of a synchronization mechanism: 1) in the application, or 2) in the transport protocol. Both have advantages and drawbacks. If the mechanism is in the application the application must handle all the synchronization itself, resulting in more application overhead, but giving flexibility and a possibility for a more complex implementation. A transport protocol implementation tries to simulate an end-to-end connection as a circuit. The application receives the same temporal relationships the sender has sent. This simplifies the application design.

1.1.1. RTSM

Real-time synchronization model (RTSM), application model suited for packet networks was presented C-C-C. Yang and J-H. Huang in [1]. The elements of RTSM are place, token and transition. The places are used to present the medium units (audio segments, video frame) and their corresponding action (playing audio, displaying video). Tokens are used to present the states inside places. A place without a token means that the place is inactive. Transitions are used for representing synchronization relationships. In order to avoid the kind of blocking as in example presented before RTSM has a mechanism called enforcing of transitions.



V:video, Tx:text, A:Audio

Figure 2: Example of figure 1 with enforced transitions used by RTSM model.

The firing rule of enforced places is that once an enforced place gets unblocked it will be immediately fired regardless of the states of other places. In figure 2 this means that if A1 becomes unblocked A2 is fired regardless of the state of V2 or Tx1. At the same time the tokens of the places Tx1 and V2 must be cleared (made inactive), since they are obsolete due to the firing of state t3. This action is called backtracking.

Definition of RSTM:.

Definition 1: RSTM is a seven-tuple {T,P,E,A,D,M}, where

$T=\{t_1, t_2,\ldots,t_n\}$	***Transitions***
$\mathbf{P=\{p_1,p_2,\ldots,p_m\}}$	Regular places, single circles
$\mathbf{E=\{e_1,e_2,\ldots e_s\}}$	Enforced places
$\mathbf{S=P\cup E}$	All places
$\mathbf{A:\{T\ X\ S\}\cup\{S_T\}}$	Directed arcs
D:S→Real number	Time duration of places
$\mathbf{R:S\rightarrow\{r_1,r_2,\ldots r_k\}}$	Type of media
$\mathbf{M:S\rightarrow\{1,2,3\}}$	State of places

Each place may be in one of the following states[1]*:*

0: no token

1: token is blocked (symbol: cross in the place)

2: token is unblocked (symbol: dot in the place)

Definition 2: To initiate RSTM is to add a token to the initial place.

Definition 3: Firing rules of RSTM

case(a): Transition t_i does not contain any enforced place in its input places.

1) Transition t_i fires immediately when each of its input places contains an unblocked token.

2) Upon firing, transition t_1 removes a token from each of its input places and adds a token to each of its output places.

3) After receiving a token, a place p_j remains in the active state for the interval specified by the duration τ_j. during this interval, the token is blocked.

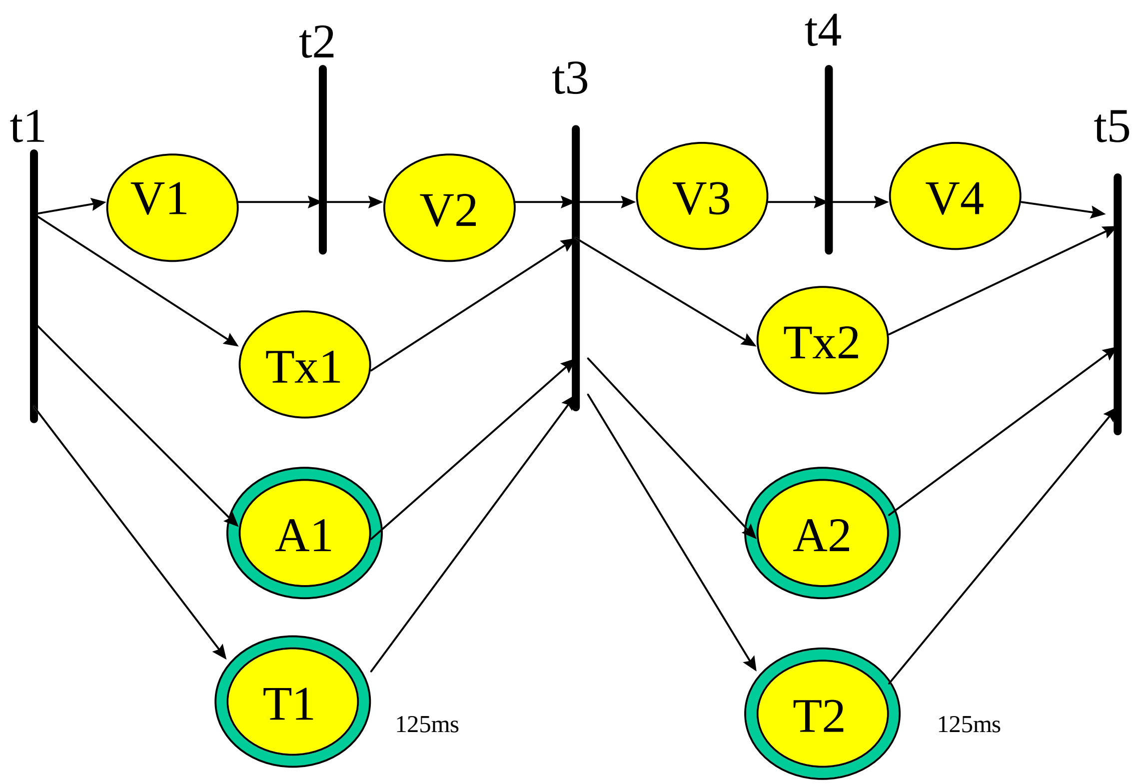
Case(b): Transition t_i contains at least one enforced place e_i in its places.

1) When the token in any of the enforced places e_i becomes unblocked, transition t_i is fired regardless of the state of other input places.

2) Upon firing, a set of backtracking rules is exercised to remove tokens from their input places. Meanwhile transition t_i adds a token to its output places.

1.1.2. Concept of Key Medium and Time Medium

A key medium is a medium that is assigned as enforced places, and will not be blocked due to late transmission. Voice was the key medium for synchronization in the previous example. The concept is similar to that of the master/slave policy proposed by Anderson, et. Al. in [2]. A time medium is an independent and virtual medium containing the deterministic duration of time specifying the real-time constraints between transitions. Time medium is needed in the case that the late transmission of the key medium. If we again consider the example of figure 2, in the case of the late transmission of the packets of A_1 it would be reasonable to fire t_3 to activate A_2, V_3 and Tx_2 instead of waiting for A_1 in order to maintain the quality of audio. By assigning the places of time medium to be enforced places, we achieve the triggering of t_3 when the duration of time specified by the time medium expires, even if the places of other media are still blocked.



V:video, Tx:text, A:Audio

1.2. Multimedia Synchronization Transport Protocol

The Multimedia Synchronization transport Protocol is an implementation of RTSM presented earlier in chapter 3.2. As such it is a protocol that provides synchronization at the session level, but no transport level services. The current implementation is a prototype designed for the evaluation of the MSTP and RSTM in comparison to other similar protocols and models (e.g. RTP).[1]

1.2.1. The Architecture of MSTP

MSTP architecture is multi-protocol based. In Figure 3 are the different entities of MSTP. The MSTP manager is responsible for the set up of all multimedia connections. In MSTP each media stream is a separate sub-connection and a sub-protocol is used for the setup with its own quality of service. The sub-protocol manages the connections at the transport layer. The MSTP manages the setup and release of each sub-protocol connection. A single MSTP manager manages the connections of a single application.

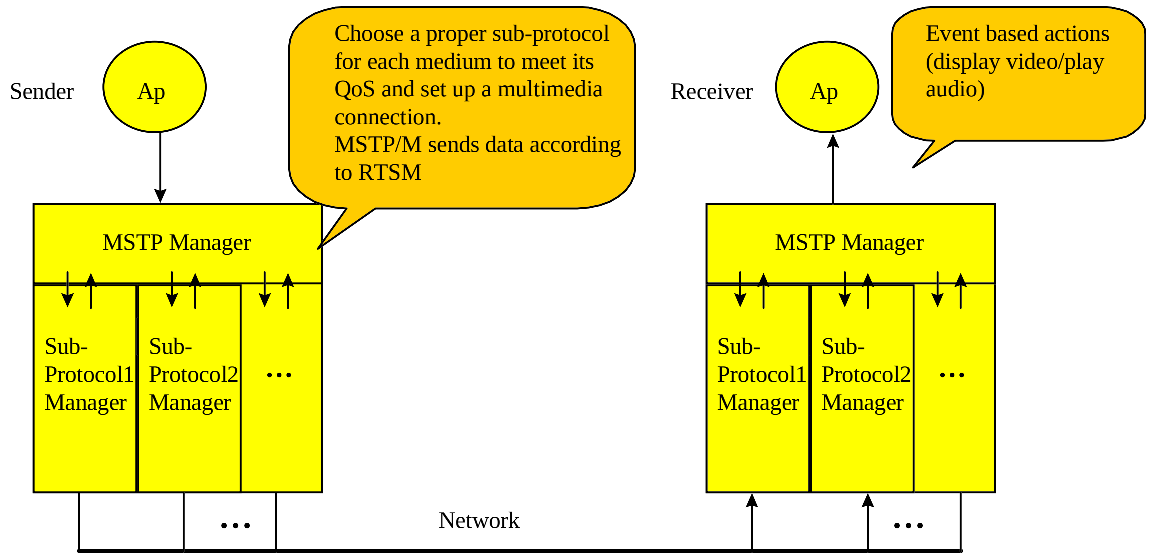


Figure 3: The architecture of MSTP.

The connection is setup as follows. The MSTP manager at the sender accepts the RTSM and QoS for each medium from the application and sets up a multimedia connection with the MSTP manager of the receiver. After setup the packets are sent according to the RTSM and the receiver MSTP deals with the out-of-sync situations and forwards the packets to the receiver application. The application development is simplified: there is no need to design a synchronization mechanism fore every application [1].

1.2.2. MSTP Packet Formats

There are two kinds of MSTP packets: data packets and control packets. The control packets are used for sending control information: to set up connections to release connections and to change the configuration of connections. The fields of the RMTP control packet are (Figure 4): PacketType which is C for control packets, KeySCID which is used to denote the key medium of the connection and the duration of the time medium. For an example application the control packet is in Figure 5. The key medium is audio the audio packet length is 125ms.

PacketType	KeySCID	TimeMediumDuration

C:control, KeySCID:Key Sub-Connection ID

Figure 4: RMTP control packet.

C	SCID for audio medium	125ms

Figure 5: An example control packet

The data packet format is in Figure 6. The SCID indicates which sub-connection the packet belongs to. The KeyRefSeq indicates which key medium packet the data packet needs to be synchronized with. The PC and TID values are used for packets of places which do not feed into the same transition with the key medium.

D	SCID	KeyRefSeq	PC	TID	SEQ	other header info	data portion

PC:place count, TID:transition ID, SEQ sequence number

Figure 6: RMTP data packet format.

The data packets for the example used in Concept of Key Medium and Time Medium[1] are

in Figure 7. The PC and TID fields are null because they feed into the same transition with the key medium.

A1	D	1	x	x	x	1	Others	Data
Tx1	D	2	1	x	x	1	Others	Data
V1	D	3	1	1	t2	1	Others	Data
V2	D	3	1	x	x	2	Others	Data
A2	D	1	x	x	x	2	Others	Data
Tx2	D	2	2	x	x	2	Others	Data
V3	D	3	2	1	t4	3	Others	Data
V4	D	3	2	x	x	4	Others	Data

SCID of audio=1, SCID of text =2, SCID of video=3, x=null

Figure 7: RMTP data packet formats.

1.2.3. Experiments with MSTP

The MSTP protocol sample implementation was tested over a WAN environment simulated on an Ethernet. A gateway process was added on the receiving hosts that will drop a packet with some probability and the forwarding of the packet is delayed. The delay is generated by a delay model. The criterion used for the evaluation of the synchronization mechanism was the delay variance between the actual playing time and the proper playing time of each packet. MSTP performance was compared against no synchronization and synchronization with Petri-net. In the measurements [1] MSTP performed well and met the goals of the design. The delay variances of both audio and video packets were kept small while still keeping the media in sync. The video late packet drop mechanism worked and the effective frame rate was reduced. This mechanism worked with jpeg-video, but does not work equally well with MPEG - video with inter-frame coding. Discarding frames of the I-frame will end up in the damage of several frames. If good video quality is needed the video will need to be assigned more bandwidth through a bandwidth reservation process and end-to-end delay guarantees (better QoS) at the network level. In addition video needs to be assigned as the key medium. Perfect quality can not be achieved unless lost packets are retransmitted or some kind of error correction scheme is used, for example forward error correction (FEC) . Retransmission is against the real-time requirements, resulting in a compromise between the real-time and error-free requirements.

1.3. IETF Real-Time Protocol

The IETF audio-video transport group started work on a real-time transport protocol in 1993. The aim of the protocol was at providing services required by interactive multimedia conferences, such as playout synchronization, demultiplexing, media identification and active-party identification. However, not only multimedia conferencing applications can benefit from RTP, but also storage of continuous data, interactive media distribution, distributed simulation, active batch, and control and measurement applications could take advantage of the possibilities RTP brings [3].

The design goals of RTP were [3]:

1. Content flexible - RTP should not be limited to only voice and video conference.
2. Extensible - RTP should be able to accommodate new services as operational experience accumulates.
3. Independent of lower layer protocols - RTP should work with UDP, TCP, ST-II and ATM.
4. Bridge/RTP gateway compatible - it should be possible to aggregate several media streams into a single stream and possibly retransmit it with a different encoding.
5. Bandwidth efficient - header overhead in short voice packets can be as much as 100%. For example with a 65ms packetization interval using 4800 bit/s encoding produces 39 byte packets. IPv4 incurs 20 bytes of headers, UDP an additional 8 bytes and the datalink layer at least an additional 8 bytes. With RTP headers around 4 to 8 bytes the total of headers is around 36 or 40 bytes per packet. This could stand in the way of running RTP over low-speed links.
6. International - a and μ law encoding as well as non US-ASCII character sets should be included
7. Processing efficient - even the longest packetization intervals give packet arrival rates of 40 per second for a single voice channel. Per packet processing overhead may become a concern.
8. Implementable now - the protocol is more or less experimental and the lifetime of the protocol was not anticipated long, so it must be implementable with the current hardware and software.

1.3.1. The Architecture of RTP

RTP concept consists of two closely linked parts: the real-time transport protocol (RTP), to carry data that has real-time properties and the RTP control protocol (RTCP), to monitor the quality of service and convey information about the participants in an on-going conference. RTP implementation will often be integrated into application processing rather than being implemented as a separate layer. The RTP framework is deliberately “loose” allowing for modifications and tailoring. In addition to RTP a complete specification for a particular application will require a payload format and a payload profile specification. A payload format defines how a particular payload (e.g. audio, video) is to be carried in RTP. A payload format specification defines how a set of payload type codes are mapped into payload formats (e.g. media encodings).

RTP-session setup consists of defining a pair of destination transport addresses one IP address plus a UDP port pair, one for RTP and one for RTCP. In the case of a multicast conference the IP address is a class D multicast address. In a multimedia session each medium is carried in a separate RTP session, with its own RTCP packets reporting the quality of that session. Usually additional media are allocated in additional port pairs and only one multicast address is used for the conference.

1.3.1.1. RTP Packets

RTP smoothes out the effects of network delay variance e.g. performs synchronization, see [4]. This is done by adjusting the playout time so that the temporal relationships between samples are restored and late arriving packets are discarded. In order to do this the RTP header is added to the continuous media sample or a group of samples.

The RTP header format is in Figure 8. The fields included are:

- V - version information, for distinguishing between different versions of RTP (2 bits).
- P - padding (1 bit), if the padding bit is set the packet contains one or more additional padding octets at the end which are not part of the payload. Padding is needed by some encryption algorithms with fixed header sizes.
- X - extension, if the extension bit is set, the fixed header is followed by exactly one header extension.
- CC - CSRC count contains the number of CSRC identifiers that follow the fixed header.
- M - marker (1 bit), the marker bit is defined by the a payload profile. It is intended to allow significant events such as frame boundaries to be marked in a packet stream.
- PT - payload type (7 bits) identifies the format of the RTP payload and determines its interpretation by the application.
- Sequence number - (16 bits) increments one for each RTP data packet sent. May be used by the receiver for detecting packet loss and restoring packet sequence.
- Timestamp - (32 bits) a media specific timestamp containing the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived form a clock that increments monotonically and linearly in time to allow synchronization and delay variance calculations.
- SSRC - (32 bits) synchronization source identifier identifies the synchronization source. The identifier is chosen randomly, so that no two sources within a session have the same SSRC identifier.

- CSRC - (0-15 x 32 bits) the contribution g source identifier identifies the contributing sources for the payload contained in the packet.

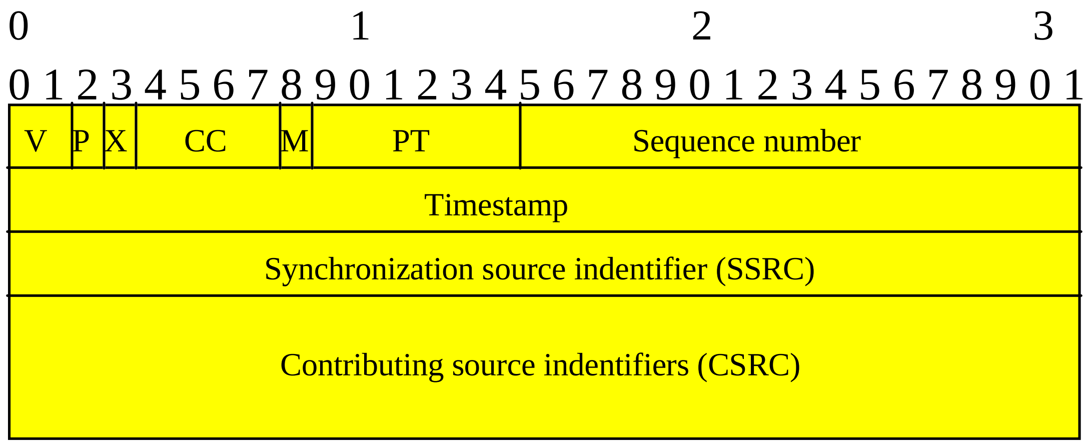


Figure 8: RTP header [4]

1.3.1.2. RTCP

RTP control protocol is based on periodic transmission of control packets to all the participants of a particular session. The control packets are distributed in the same way as the data packets. RTCP has four separate functions:

1) The primary function is to provide feedback on the quality of the data distribution. The feedback can be used to control adaptive encoding. Experiments with IP multicasting have shown that feedback is also critical for diagnosing faults in the distribution. The feedback function is achieved with sender (Figure 9) and receiver (Figure 10) reports.

2) RTCP keeps track of all participants of a session. It does this by carrying a transport level identifier of each source called the canonical name (CNAME) and the synchronization source identifier. The SSRC may change in a session. The CNAME is also needed for the synchronization of multiple related streams (audio and video).

3) RTCP packets are sent in order to perform functions 1 and 2, therefore the rate at which RTCP packets are sent must also be controlled. This rate controlling is done by RTCP. The number of participants observed is used for determining the rate at which packets are sent. The more participants there are in a conference the less frequently each participant sends packets.

4) The forth optional function is to carry minimal session control information, for example participant identification to be displayed in the user interface.

Functions 1-3 are mandatory when RTP is used in an IP multicast environment, and are recommended in all environments.

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

V | P | X | RC | M | PT=SR | Length	Header
Source indentifier	
NTP timestamp, most significant word	Sender info
NTP timestamp. least significant word	
RTP timestamp	
Sender's packet count	
Sender's octet count	
SSRC_1 (SSRC of first source)	Report block 1
Fraction lost | Cumulative number of packets lost	
Extended highest sequence number received	
Interarrival jitter	
Last SR (LSR)	
Delay since last SR (DLSR)	
SSRC_2 (SSRC of second source)	Report block 2
• • •	
Profile specific extensions)	

Figure 9: Sender report RTCP packet [4]

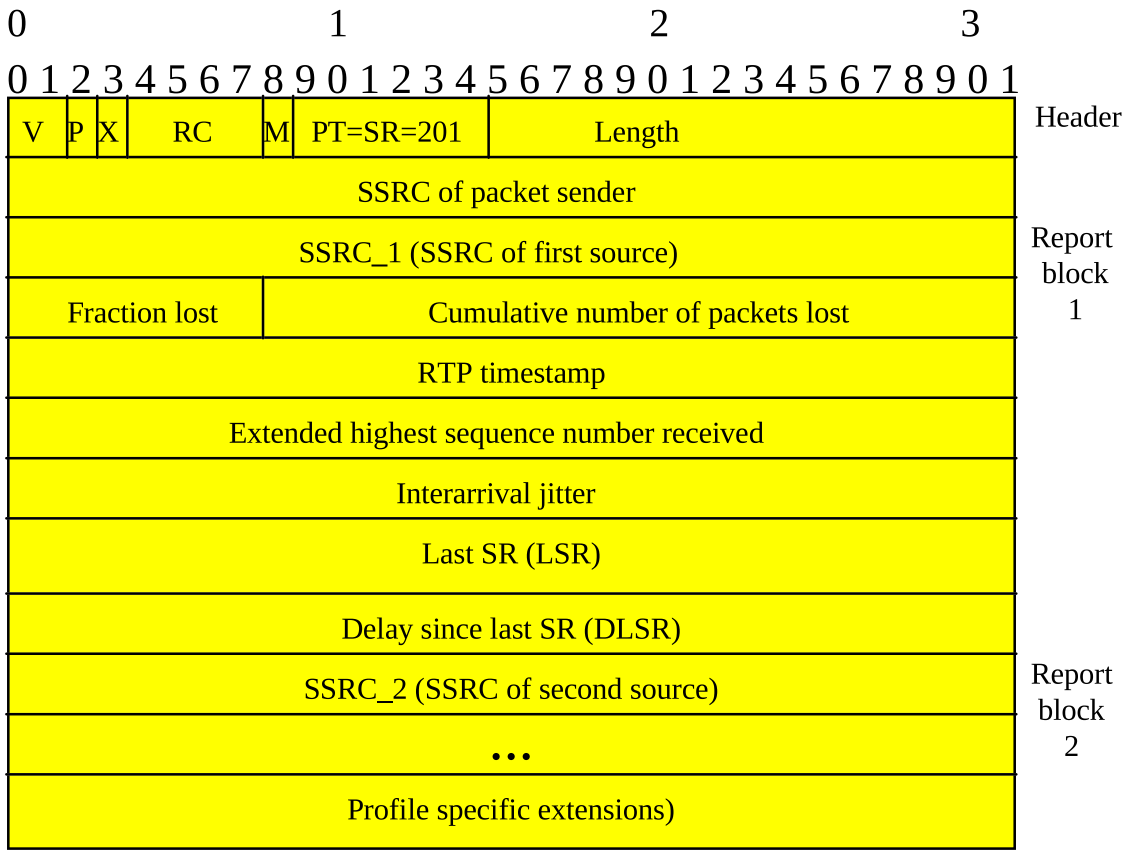


Figure 10: Receiver report packet [4]

1.4. H.225.0

ITU-T H.32x recommendations define visual telephone terminals and how to run the terminals over various networks. H320 applies to N-ISDN while H.321 applies to B-ISDN (ATM). H.322 and H.323 apply to LANs. The difference between the latter two is in that H.323 applies to LANs without QoS guarantees and H.322 to LANs with QoS guarantees. In the scope of media delay variance and synchronization the recommendation H.323 is interesting and in particular H.225.0. Most of this section is from the ITU-T recommendation H.225.0 [5].

The scope of recommendation H.323 is in Figure 11. H.225.0 covers protocols and message formats. It is designed to operate over various LANs, such as IEEE 802.3 and IEEE 802.5. It acts as an convergence layer, above the transport layer. H.225.0 is protocol independent and can be used over LANs with QoS guarantees as well. *The scope of H.225.0 is communication between H.323 terminals and gateways in the same LAN using the same transport protocol*. H.225.0 may be used over interconnected LANs or even over the Internet, but the performance is acceptable only when the network load is low. The scope of H.225.0 is in Figure 12.

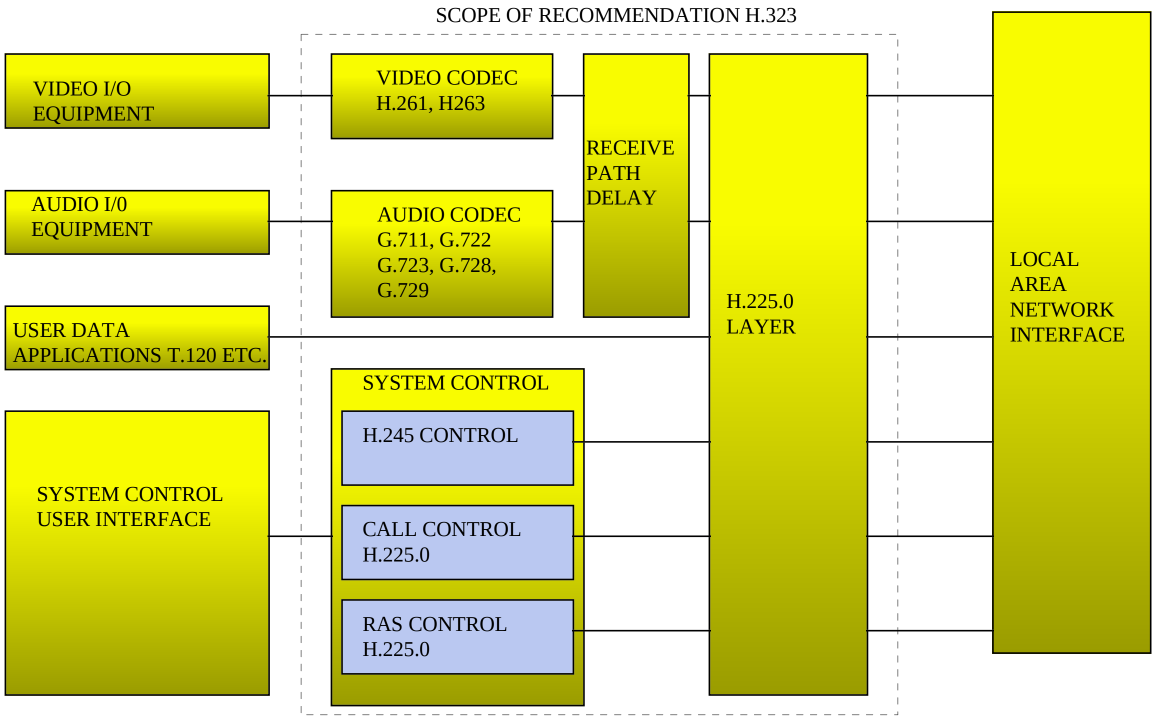


Figure 11: H.323 scope

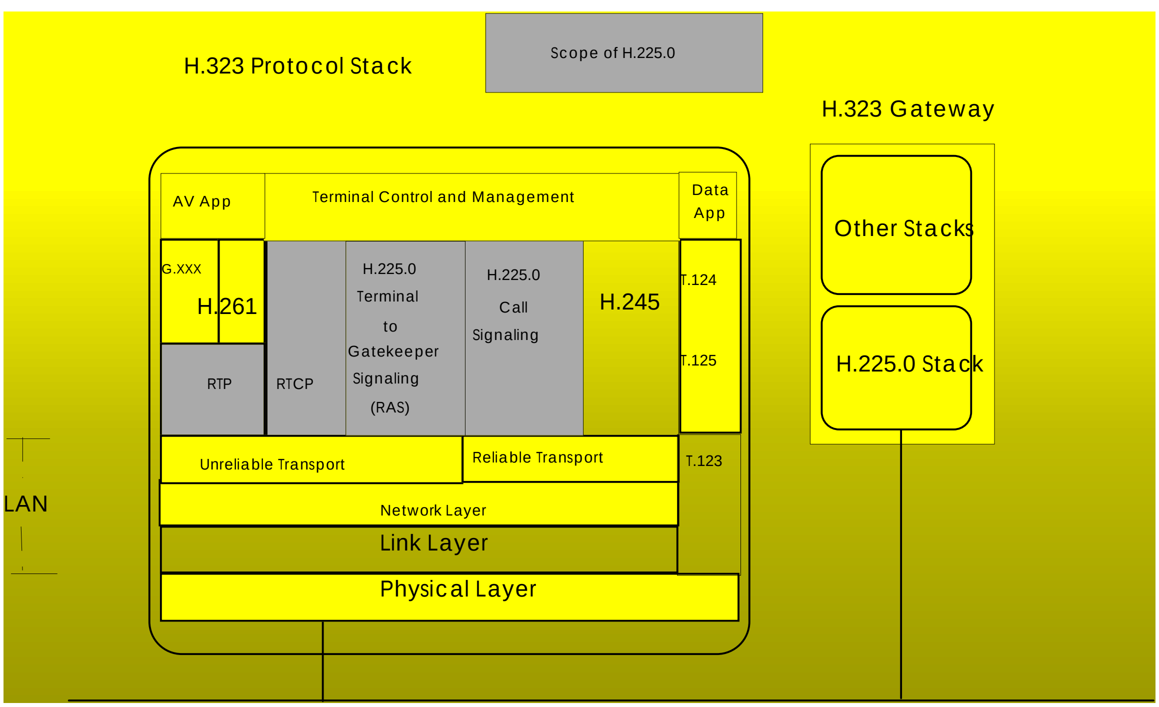


Figure 12:The scope of H.225.0

1.4.1. Use of RTCP in Measuring QoS

RTCP sender reports are used for three main purposes: to allow synchronization of multiple RTP streams, to allow the receiver to know the expected data and packet rates and to measure the distance in time to the sender. The most important issue for H.225.0 is that of the synchronization of multiple sources.

The receiver reports are used to measure the QoS of the connection: fraction lost, cumulative packets lost, the extended highest sequence number received and the inter-arrival delay variance. The cumulative number of packets lost and the sequence number are used to compute the packets lost since the last receiver report. This can be used for determining the long term congestion in a LAN. If the state of congestion is higher than the value set by the terminal manufacturer (programmer) then the terminal should reduce the media rate. High interarrival delay variance and long intervals in sender reports can also be used as indicators of high state of congestion.

As an option H.245 level signaling can be used in reducing delay variance and related delays.

1.4.2. Procedures for maintaining QoS

The methods that can be used by H.323 terminals/gateways to respond to congestion can be grouped in two: those that respond to short term problems and those that respond to longer term problems. The methods do not seek to maintain QoS, but instead to provide an orderly degradation of service. Short term responses are responses to problems like lost or delayed packets. A typical long-term response would be that of the growing congestion on the LAN. The media degradation order is: video, data, audio, control.

There are three typical short-term responses: reducing the frame rate for a short period of time, reducing packet rate by mixing audio and video in same packet and packet rate reduction by video fragmentation at the H.261 macro block level. More sophisticated responses would be increasing the amount of redundancy information in packets or increasing the amount of information used in Forward Error Correction [6, 7, 8].

Long-term responses are: reduction of media bit rate, turning of media of lesser importance and returning a busy signal to the receiver as indication of LAN congestion. The busy signal sending can be combined with turning of media.

In a multi-router configuration reacting to delay variance can be difficult. It may be impossible to distinguish the source of delay variance when there is a lot router incurred reordering and varying delay of packets.

2. Beyond Synchronization - Reservations, QoS Guarantees and Packet Scheduling

In this chapter we will go through some Internet related protocols that aim to provide better quality of service for applications through resource reservation mechanisms. We will first introduce Internet stream protocol version 2+. This will be followed by the Tenet Suite. The chapter ends with a brief presentation of the Internet Engineering Task Force's Integrated Services model and the associated resource reservation setup protocol, RSVP.

2.1. Internet Stream Protocol Version 2+(ST-II+)

Internet stream transport protocol was the first attempt to provide some kind of quality of service (QoS) in a packet network environment. It was published already in the late 1970's. and was used in experimental voice and video transmission . Later building on the experiences with ST IETF developed ST-II a revised version of the stream protocol in 1990 [9]. In 1993 IETF started a new working group on ST-II as a part of the ongoing efforts to develop protocols that support resource reservation. The mission of the new group was to clean up existing specifications and to ensure better interoperability between implementations. There existed over 15 different implementations of ST-II all mutually incompatible. The use of ST-II+ is experimental and real-world applications will be most likely based on the combination ATM/RSVP/IP/RTP. Still there are some very fine ideas behind the ST specification, and a brief introduction is needed to get the right perspective on the current resource reservation and real-time protocol issues. Some of the ideas behind the newer protocols were already presented in the earliest draft of ST in late 1970's.

ST-II is an experimental connection-oriented interworking protocol operating at the same layer as connectionless IP. The role of ST-2 is complementary to IP, adjunct to IP not a replacement. The main applications for ST-2 were thought of as the transport of multimedia, e.g. digital audio and video packet streams, and distributed simulation and gaming across intranets [10].

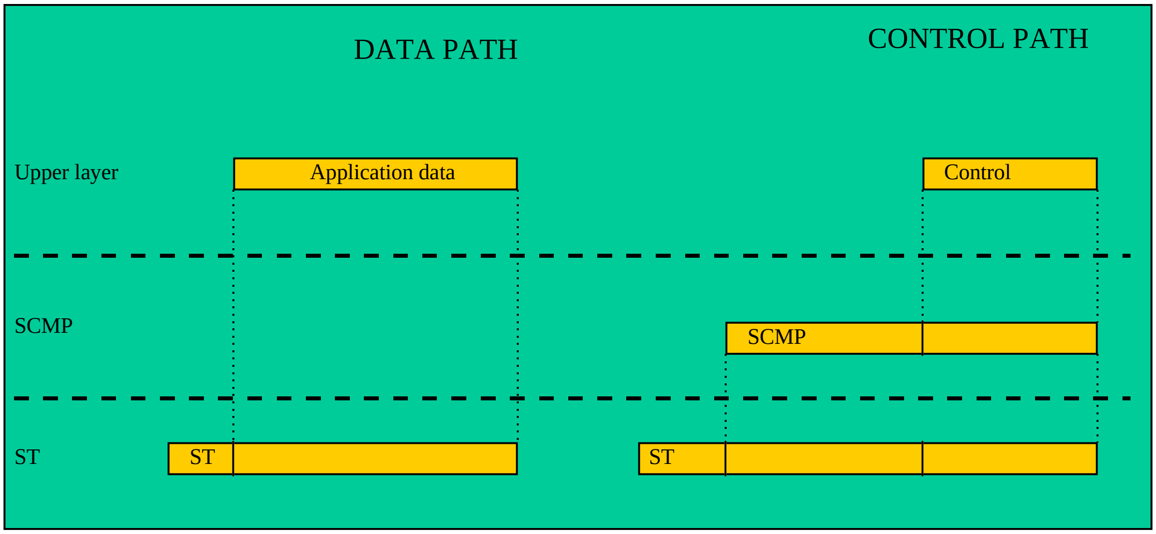


Figure 13: ST-2 data and control path

ST-2 can be used for bandwidth reservation for real-time connections across network routes. Quality of Service guarantees are achieved in addition to the reservation mechanism, with network access and packet scheduling. ST-2 ensures that real-time packets are delivered within their deadlines. This facilitates the smooth playout of time-critical media, in a way that the traditional best-effort IP can not provide [**7**].

ST consist of two protocols just like IP: ST for data transport and another protocol, Stream Control Message Protocol (SCMP) for all control functions. ST is simple and contains only one data format, thus achieving fast and efficient data forwarding and the goal of low communication delays. SCMP on the other hand is very complex, offering more functionality than ICMP in IP. The SCMP packets are encapsulated in ST packets in the same manner as ICMP packets are transferred within IP packets, see Figure 13.

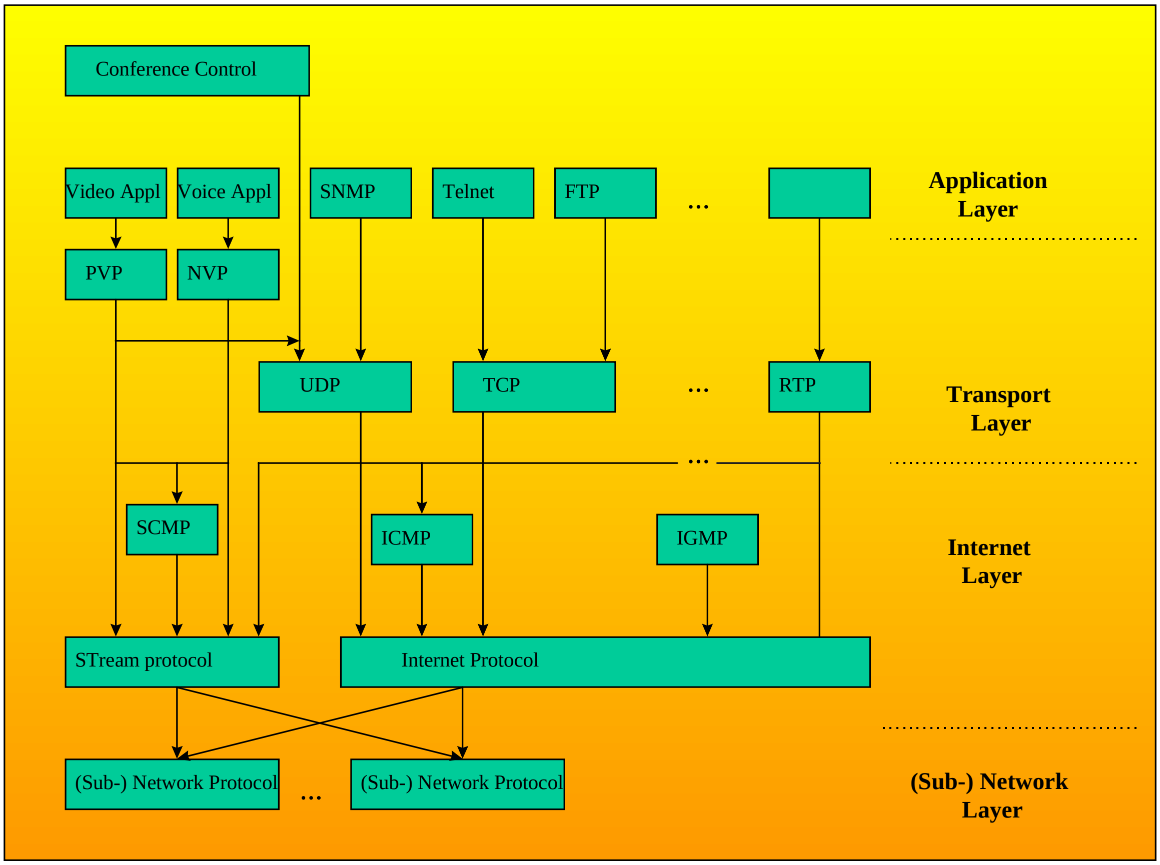


Figure 14: Protocol relationships.

The position of ST-2 in relation to other IP-family protocols is represented in Figure 14. ST-II is designed to coexist with IP in each node. A multimedia application could use ST-II for the transfer of real-time data and IP for the transfer of data and control information. IP is accessed with TCP or UDP and ST-II via new end-to-end real-time protocols.

ST-II and IP use the same addressing schemes to identify different hosts. The internetworking number of ST-II is 5, while IP uses number 4. ST-II is a network layer protocol and operates independent of its underlying subnets. ST-II messages can be encapsulated in IP packets if needed.

Possible transport layer protocols on top of ST-II are RTP and possibly application specific ones (ST-II was designed to be an experimental protocol) like those used in the older versions of the Internet conferencing tools NV, VIC and VAT. The earliest real-time transport layer protocols were Network Voice Protocol (designed in the seventies) and Packet Voice Protocol (designed in the eighties). They have been used on top of ST-II in experiments, but RTP has replaced them.

2.1.1. The Architecture

2.1.1.1. Data PDU

The data transfer protocol defines the format of the data packets belonging to a stream. Data packets are delivered to the targets through previously established stream paths. The stream paths are setup by the setup protocol. Data packets are delivered with the quality of service associated with the stream.

In Figure 15 is the ST PDU used between ST agents. The header can encapsulate either a higher layer PDU or an ST control message. The D bit is used for distinguishing between the two. Packets having a non-zero D-bit are data packets and their interpretation is left for the higher layer protocols. Other fields in the header are the ST version number, a total length field, a unique ID, and the stream origin 32-bit address.

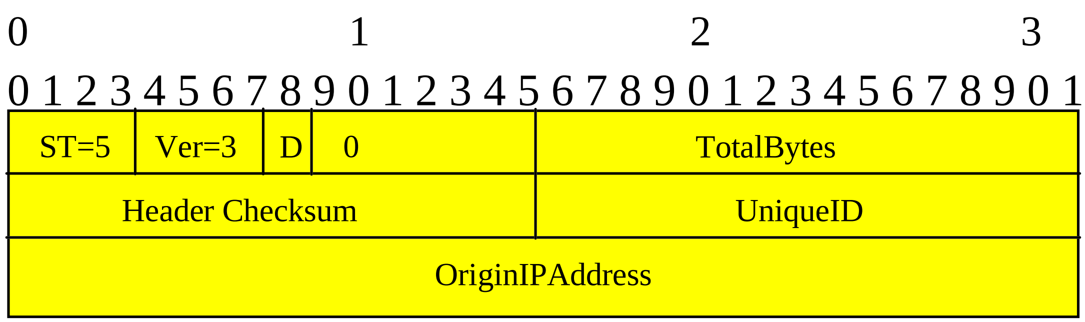


Figure 15: ST header.

2.1.1.2. ST-II+ Flow Spec and resource reservation

The quality of service parameters for a stream are negotiated by the SCMP as a part of the connection establishment. These parameters form the ST-II FlowSpec associated with each stream. The entities participating in the quality of service negotiation are: 1) the application entities on the origin and target as the service users, 2) ST agents, and 3) local resource managers, LRMs. The origin application supplies the initial FlowSpec requesting particular service quality. The ST agent obtaining the FlowSpec as a part of connection establishment message presents it to the LRM. The information a ST agent relays includes: the flow specification with the desired quality of service for the stream, the version number associated with the stream and the information groups the stream is a member of.

ST-2 is not dependent of any particular format, it is expected that other versions of the flow specification will be needed. Different flow specifications are distinguished by the version field number. A single stream is always associated with a single flow specification format. In

Table 1 are the currently defined version field values of ST-2 [10].

Table 1: Flow Specification version field values.

#	*Version*	*Comments*
0	Null FlowSpec	must be supported
1	ST Version 1	
2	ST Version 1.5	
3	RFC 1190 FlowSpec	
4	HeiTS FlowSpec	
5	BerKom FlowSpec	
6	RFC 1363 FlowSpec	
7	ST2+ FlowSpec	must be supported

The ST-2+ flow specification format is in Figure 16. The real-time requirements are defined by a QoS class, precedence and three QoS parameters: message size, message rate and end-to-end delay. The QoS class indicates what kind of QoS guarantees are expected by the application, e.g. predictive or strict. The QoS parameters are expressed via a set of values:

> *desired* - the QoS desired by the application,
>
> *limit* - the lowest QoS accepted by the application
>
> *actual* - indicate the QoS the system is able to provide.

There are currently two defined QoS classes:

> QOS_PREDICTIVE - the predictive class implies that the negotiated QoS may be violated for short time intervals during transfer. The resource reservations are made for the normal (or average rate), not the peak rate.
>
> QOS_GUARANTEED - the guaranteed class implies that the negotiated QoS of the stream is never violated during transfer. The application provides values that take into account the worst possible case, e.g. the desired rate is the peak rate of the transmission. The resource reservations are made for this peak rate. This lead to overbooking of resources, but provides strict real-time guarantees. QOS_GUARANTEED is optional.

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

QosClass	Precedence	0 (unused)
DesRate		
LimitRate		
ActRate		
DesMaxSize		LimitMaxSize
ActMaxSize		DesMaxDelay
LimitMaxDelay		ActMaxDelay
DesMaxDelayRange		ActMinDelay

Figure 16: ST-2+ FlowSpec format.

2.1.1.3. Control PDU

SCMP control messages are exchanged by the neighbor ST agents using a D-bit of zero. The control protocol expects a response for all requests. The format of all control messages is in Figure 17.

0 1 2

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

OpCode	Options	TotalBytes
Reference		LnkReference
SenderIPAddress		
Checksum		ReasonCode

OpCodeSpecificData

Figure 17: ST control message format.

2.2. Tenet Suite 2

Work on the first "complete" real-time protocol suite was begun in 1987 in the Berkeley Computer Science Institute, during the course of studies in operating system support for multimedia applications. The assumption then made was that a set of performance guarantees were needed in future networks, and that the networks would be based on packet switching. The idea was to design a set of algorithms that would offer a packet-switched network's real-time clients the desired guarantees by providing real-time channels. The scheme is based on: the real-time abstraction itself, admission control, connection-oriented communication, channel rate control and deadline-based scheduling for real-time packets. The first version had several limitations, such as the support for only unicast channels and the lack of routing algorithms for real-time channels. The suite was finished 1991. The second Tenet Suite 2 work started in 1992. Suite 2 is based on Suite 1, but it has a more flexible client side interface and multi-party communication support (multicasting).

2.2.1. Architecture of the Tenet Suite

The architecture of the Tenet Suite is presented in Figure 18. As presented by [11], the main principles behind the architecture are:

1. All layers in a network's architecture, in particular the datalink layer, must be capable of supporting guaranteed performance services. Examples of datalink layers meeting these requirements are synchronous FDDI, ATM with a suitable signaling protocol and admission tests and 100VGAnyLAN. Ethernet IEEE 802.3 can not offer delay guarantees. Performance bounds can be offered to applications by implementing real-time protocols at the networks and transport layers. The Tenet network layer is called the Real-Time Internetwork Protocol (RTIP) and the transport layer implementation consists of the Real-time Message Transport Protocol (RMTP) and the Continuous Media Transport Protocol (CMTP).
2. In the Tenet scheme real-time channels are set up in an establishment phase that precedes data transfer. Admission control tests are run in each node along the path of the real-time channel and must succeed before the data transfer can begin. The control functions are provided by the Real-time channel administration protocol.
3. Real-time applications have requirements that can be expressed with general performance or reliability parameters. This eliminates the need for media-specific protocols at the network and transport layers.
4. The design goal of the Tenet suite is to offer real-time service in integrated-services network. Hence it has been designed to operate in coexistence with traditional TCP/IP and UDP/IP and their real-time traffic.

5. A means of detecting and recovering from failures is needed. The connection management is provided by RCAP and the Real-Time Control Message Protocol, RTCMP.

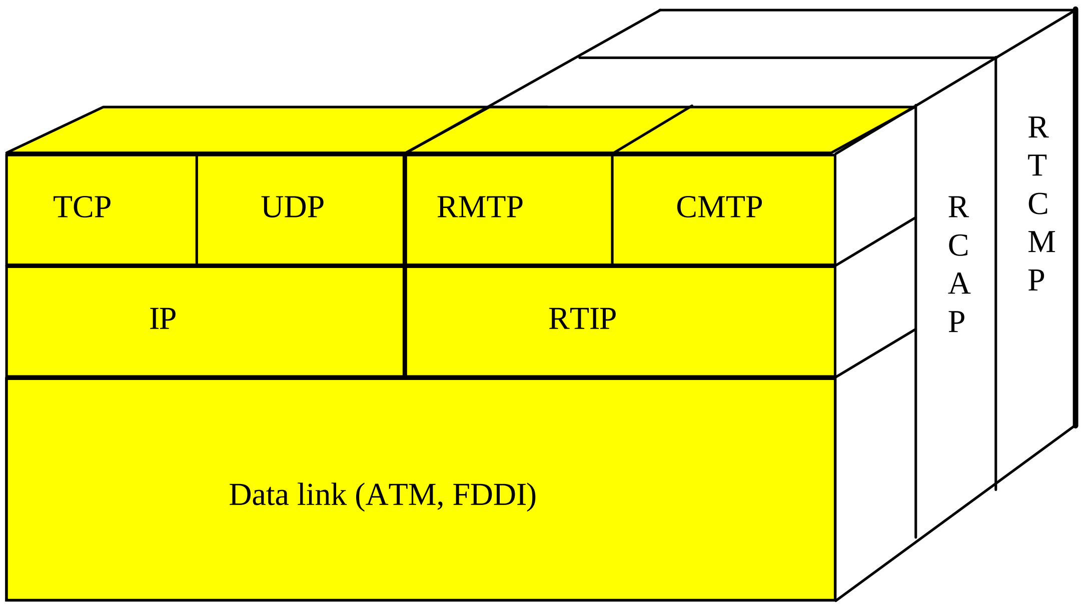


Figure 18: The Tenet real-time protocol suite.

2.2.2. Design of the Tenet Protocols

The Tenet Suite consists of five separate entities RMTP/CMTP, RTIP, RCAP and RTCMP. Here I will go through all of these briefly.

The signaling and control functions are provided by the real-time channel administration protocol RCAP. The main duties of RCAP are real-time channel setup and teardown and channel status reporting. A list of RCAP messages is in Table 2. Each of the messages has a implicit path of propagation, either down- or upstream. Channel setup is performed in a single round trip.

Table 2: RCAP messages

Message Name	*Direction*	*Description*
establish_request	Downstream	Request to establish a new channel.
Establish_accept	Upstream	Indicates acceptance of a new channel.
Establish_denied	Upstream	Indicates that a channel establishment request was rejected.
status_request	Downstream	Request channel status at each node.
status_report	Upstream	Returns data collected by a status_request.
close_request_forward	Downstream	Message from source to close a channel.
close_request_reverse	Upstream	Message from destination to close a channel.

During the channel establishment the client must specify the QoS requirements and a worst-case description of the traffic it will generate. The QoS parameters are presented in Table 3 and the traffic parameters are in Table 4. The channel establishment can be seen as signing a contract: the network guarantees the performance bounds requested by the application. The contact is valid until the channel is torn down. RCAP does not offer re-negotiation of an established channel.

Table 3: QoS parameters.

Symbol	*Description*
$\mathbf{D_{max}}$	Upper bound on end-to-end message delay.
$\mathbf{Z_{min}}$	Lower bound on probability of timely delivery.
$\mathbf{J_{max}}$	Upper bound on delay variance.
$\mathbf{W_{min}}$	Lower bound on probability of no loss to buffer overflow

Table 4: Traffic parameters.

Symbol	*Description*
$\mathbf{X_{min}}$	Minimum inter-message time
$\mathbf{X_{ave}}$	Minimum average inter-message time
I	Averaging interval
$\mathbf{S_{max}}$	Maximum message size

The network layer protocol of the Tenet Suite is RTIP, the real-time Internet protocol. RTIP is a connection oriented and its main function is to deliver packets to meet the real-time channels’ real-time requirements. RTIP data forwarding is unreliable. Data may be lost due to buffer overflow. RTIP performs rate control, delay variance control based on the QoS parameters. RTIP does not perform packet reordering. All packets are assumed to follow the same path.

RTIP header is shown in Figure 19. The header is simple and of fixed length. The first four bits are used to identify the packet as an RTIP packet in order to maintain compatibility with IP. The other fields in the packet are the version field, local channel ID, packet length field, a packet sequence number used for message reassembly, and a timestamp which indicates the time the packet was received by the RTIP module of the host machine. The timestamp is used for delay variance control.

0 3 7 15 23 31

RTIP	Version	unused	Local ID
Packet Length		Packet Sequence Number	
Timestamp			
reserved		Header Checksum	

Figure 19: RTIP packet header

The relationship between RMTP/RTIP is similar as that of TCP/IP. RMTP is lighter than TCP and it does not provide reliable service (no retransmission) [11]. The main service of RMTP is fragmentation and reassembly. RMTP header is shown in Figure 20.

0 7 15 23 31

Timestamp		
Message Length	Message Seq. #	Flags
Checksum		

Figure 20: RMTP header.

2.2.3. Experiments with the Tenet Suite

The real world is a lot more complex than the one that can be simulated. The actual behavior of workstations affect the packet service times which in turn need to be known for the implementation of admission control schemes. Therefore a real world experiment is crucial for all protocols.

The implementation tested by the Tenet Group included a number of service disciplines: delay-earliest-due-date, jitter-earliest-due-date and rate controlled static priority. The last two were implemented with both rate-jitter and delay-jitter controlling regulators. The implementation was made in a modular fashion so that other researchers could include their own scheduling disciplines.

The experiments conducted by The Tenet Group revealed the strength of the RMTP/RTIP. The effective throughput achieved was comparable to raw IP, and thus higher than that of UDP/IP. The protocol processing time in a workstation was almost independent of packet size and the driver software processing time was fixed. The rate control behavior worked as planned forcing adherence to traffic specification.

Tenet Suite was also tested in an experimental wide-area network the Sequoia 2000. The experiments there showed that RMTP/RTIP traffic was able to perform well with non-real-time load and multiple streams of RMTP/RTIP on the same link. The throughputs remained almost constant even with a substantial amount of load introduced to the network. This is mainly due to the good traffic prioritization and well behaving admission control. The quality of video over RMTP/RTIP was also compared in subjective tests against UDP/IP video. With a 99% confidence interval the mean opinion score (MOS) of RMTP/RTIP video session did not change during the experiment. With the same confidence interval the MOS of UDP/IP dropped by 54% under the same load[11].

2.3. The Internet Integrated Services Architecture

The Internet Integrated Services Architecture is the IETF's approach for providing Internet Quality of Service. The goal was provide real-time services simultaneously with the traditional non-real-time best-effort service in IP networks. The IIS is often mistaken to be the same as RSVP. The big picture includes many other elements, and is more a broad reference model, than just the RSVP.

2.3.1. The Integrated Services Model - the Core Service Model

The core service model of Internet Integrated Services, IIS centers around the question of time-of-delivery of packets. The QoS commitments made by the network are related to per-packet delay, bounds on minimum and maximum delays being the sufficient parameters. Applications are grouped in two: real-time applications and elastic applications. Elastic applications always wait for data to arrive where as real-time applications are time sensitive to data delivery in the sense that the value of the data is related to the time of arrival of the data. Real-time applications can be further grouped in two: 1) applications that need perfectly reliable upper bounds for delay 2) and applications that can tolerate and adopt to variations in delay and do not need perfectly reliable upper bounds for delay. The service model for the intolerant applications is called *guaranteed service*. The service model for the tolerant applications is called *predictive service* [12, 13].

The predictive service gives the applications a fairly, but not perfectly reliable bound for delay which can be calculated with properly conservative predictions of the behavior of other flows. The service also tries to minimize the ex post maximum delay. It does not try to minimize the delay of every packet, but rather it tries to pull in the tail of the delay distribution.

Applications can also adopt to changes in the state of congestion of the network by changing their bit rate and thus their traffic characterization. For example video conferencing application can easily change the coding scheme and reduce the frame rate.

The service model for elastic applications is called best-effort service. Also the terms ASAP (as soon as possible) and datagram service are used. Elastic applications are sensitive to delay - excess delay often shows in poor application performance. However the performance of the applications is more dependent on the average delay than on the delay distribution.

The same scheme that is used for predictive service can provide controlled link sharing. The objective is not to bound delay, but to limit overload shares of the link - thus giving the name for the new service *fair share or controlled load.* The technology behind this service and the other new services is WFQ - Weighted Fair Queuing. WFQ-scheduling in the way it is used in controlled load service is available in commercial routers today and is used to segregate traffic into classes based on things like protocol type or application..

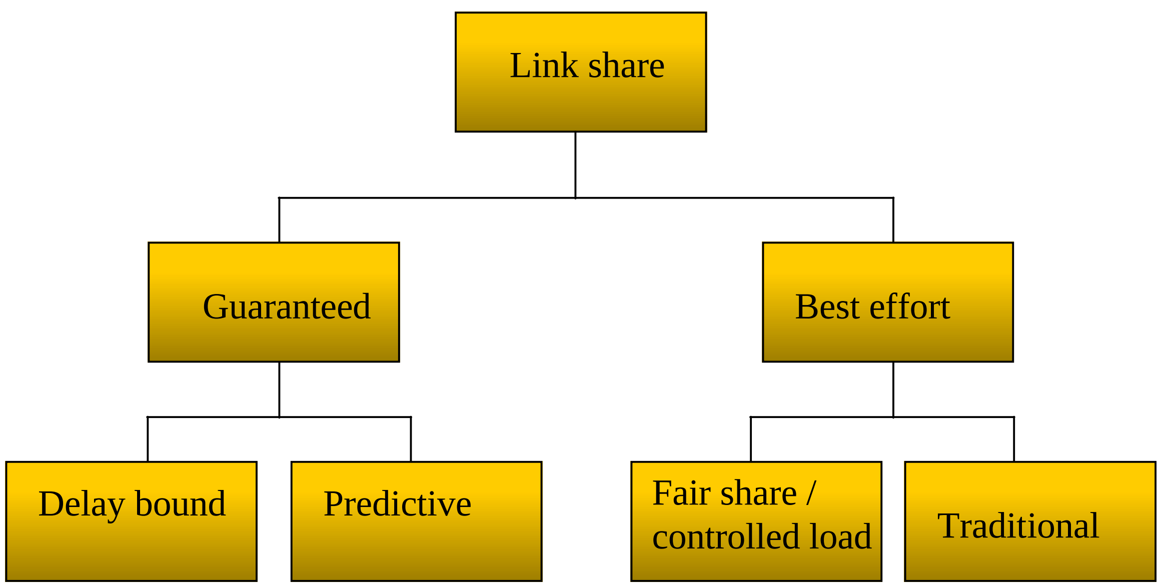


Figure 21: IETF traffic service class hierarchy [14].

2.3.2. Unified Packet Scheduling

The role of the packet scheduler in IIS is to manage the forwarding of different packet streams using a set of queues and timers. The packet scheduler is implemented at the point where packets are queued, which is typically the output driver of a typical operating system and corresponds to the link layer protocol [15].

The unified scheduling algorithm presented in [23] implements the IS model. It handles the service commitments of predicted, guaranteed and datagram service. The basic idea of the scheduling algorithm is to isolate the traffic of guaranteed service from that of the predicted service class, and to isolate guaranteed flows from each other. This is done by using the time-stamp based WFQ-algorithm. Each guaranteed service client α has a separate WFQ flow with some clock rate r^{α}. All the predicted and datagram traffic is assigned to one pseudo WFQ flow (for example flow 0). Then at each link $r^0 = \mu - \Sigma_{\alpha} r^{\alpha}$, where μ is the link speed. Inside the flow 0 there are a number of strict priority classes. The FIFO+ algorithm is used inside each priority class. Each predictive service flow is assigned a priority level at each switch and the algorithm is completely defined.

2.3.3. Classifier

The classifier maps each packet going to the scheduler onto some traffic class. All the packets in the same class get equal treatment from the scheduler. The choice of class can be based on packet headers or on a classification number on each packet. A class can also correspond to a broader category of flows. For example all video streams may be classed as one flow type and audio streams as another. A class is an abstraction and may hold only locally or it may hold for an entire backbone network. In other words routers may also map flows to classes independently. An example of packet classification is the one used by Resource Reservation Protocol (RSVP) presented in 2.3.5.

2.3.4. Admission Control: Guaranteed and Predicted Service

A fundamental aspect of the provision of delay bounds is the traffic characterization and admission control. The traffic of the source must be characterized to the network and the network must decide whether or not it can accommodate additional flows. This requires that the routers of the network understand the demands that are made on its assets at every moment. This can be achieved in numerous ways. The router can keep a table of past service requirements and make a computation based on the worst-case bounds of each service. One solution is presented by D. Clark in his paper presenting a scheme that adds service discrimination to IP-networks [23].

This scheme applies two criteria when trying to decide whether or not to admit a flow into the network.

1. Only 90% of the bandwidth should be reserved for real-time traffic and at least 10% should be left for datagram traffic.
2. Additional flows should not increase any of the predicted delay over the bounds D_i.

An example given by D. Clark in [23]:

Let ν denote the measured post facto bound on utilization on a link due to real-time traffic. Let d_i denote the measured maximal delay of the traffic in class i, and let μ denote the link speed. The admission control *criterion is that a flow promising to conform to a token bucket traffic filter (r,b) can be admitted to priority level I if*

1) $r + \nu < 0.9\mu$ *and*

2) $b < (D_j - \mathbf{d_j})(\mu - \nu - r)$*, for each class j which is lower to than or equal in priority to level i.*

A guaranteed service commitment is considered to be higher in priority than all levels *i*. The first condition guarantees that there is always 10% of the bandwidth left for datagram traffic. The second condition ensures that an admitted flow does not violate the bounds D_j even if it aggregates worst case traffic. This scenario performs well if the measures ν and $\mathbf{d_j}$ are consistently conservative estimates, not just averages.

2.3.5. Resource Reservation: RSVP

In order for the resources to be properly reserved the network has to give a characterization of the quality of service it will deliver to the source application, and the source has to characterize the traffic it will feed the network. This and the reservations for the resources are done in the IS architecture with the reservation setup protocol RSVP. There are a number of requirements for the protocol. It must be fundamentally designed for a multicast environment, and it must accommodate a vast amount of services. It must also give flexible control over sharing of resources in branches of multicast trees. It must also be robust and scale well to large user groups, and it must provide for advanced reservation of resources.

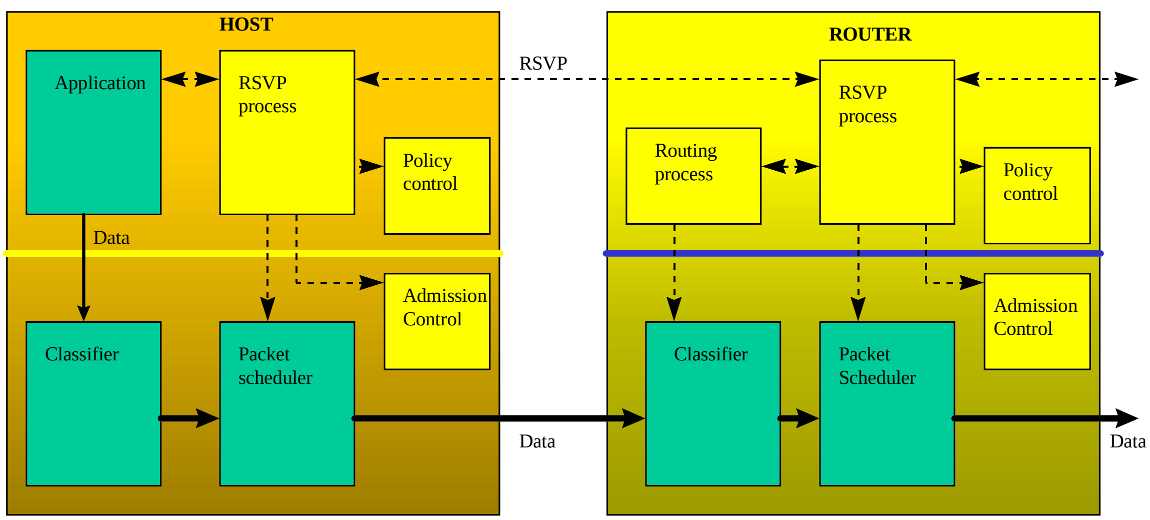


Figure 22: RSVP in Hosts and Routers

2.3.5.1. Reservation Model

A RSVP reservation request, the flow descriptor consists of a FlowSpec and a FilterSpec. The FlowSpec is used to define the desired QoS and it sets the parameters in the packet scheduler. The filter specification defines the flow that is to receive the QoS set by the FlowSpec. The FilterSpec communicates with the packet classifier. Packets that do not match any filter specifications are handled as best-effort traffic. The flow specification includes a service class and two sets of numeric parameters: Rspec defines the desired QoS and Tspec describes the flow type. The formats for the two are defined in [16].

2.3.5.2. Reservation Styles

The reservation style options are summed by Braden, et. Al. in [17]. The reservations in a session can be established as distinct for each upstream sender or they can be shared among all packets of selected senders. The selection of senders can be explicit or wildcard. An explicit reservation lists all senders and a wildcard selection implicitly selects all senders of a particular session. [17]

Table 5: Reservation styles and attributes.

Sender selection	***Distinct reservations***	***Shared reservations***
Explicit	Fixed-Filter (FF) style	Shared-Explicit (SE) style
Wildcard	non	Wildcard-Filter (WF) syle

2.3.5.3. Soft State

There are two alternative approaches to maintaining the states of connections in routers. The connection-oriented approach is called the hard state (HS). In HS the connection is created and torn down in a fully deterministic manner, so that the network is responsible for creating, keeping and destroying the necessary state. An example of the connection oriented approach are ST-2 (in 2.1), Tenet Suite 2 and native ATM. HS requires the network to be reliable. RSVP takes the soft state (SS) approach. In SS information about the reservation states are cached in the routers and periodically refreshed by the end hosts. Unused state is timed out by the routers. In the case of route change the refresh messages automatically install the state along a new route.

2.3.5.4. RSVP , Routing and QoS Routing

RSVP is not in itself a routing protocol: it is designed to operate with unicast and multicast routing protocols. An RSVP daemon consults the local routing database(s) to obtain routes. In the multicast case RSVP sends IGMP messages to join a multicast group and RSVP messages to reserve resources along the delivery paths of the group. Routing protocols handle the forwarding of the data packets. RSVP is only concerned with the QoS of the packets forwarded.

The goal of the IS group has been to extend the Internet architecture, not to replace it. It has been suggested that reservation requires route setup i.e., the imposition of a virtual-circuit Internet layer, which would clearly mean abandoning the connectionless Internet layer. The approach IS is taking is to modify the datagram forwarding of the present Internet to accommodate RSVP. Therefore there are four fundamental routing issues that need to be solved:

1) finding a route that supports resource reservation

2) finding a route that has sufficient unreserved capacity for a new flow

3) adopting to route failure

4) adopting to route changes

RSVP utilizes PATH messages to notify the receivers of a session and the traffic characteristics at the sender, and to establish the flow’s path state in the routers. With QoS routing this is not useful, since the flow path is computed based only on the receivers’ reservation. There are ways to tackle this and other problems between RSVP and QoS routing, but the fact remains that RSVP signaling and QoS based routing do not fit together well [18].

2.3.6. IIS over ATM

ATM has rapidly become **the** link layer technology. ATM can provide point-to-point and point-to-multipoint connections using Virtual Circuits (VC) with a specified Quality of Service. The leaf nodes of point-to-multipoint distribution trees can be set up and removed from the VC dynamically, thus allowing IP multicast groups to be set up. Currently ATM is the number one choice of link layer for the IIS model and RSVP.

The traditional way of handling IP traffic over ATM has been to use ether Classical IP over ATM or LAN Emulation (LANE). In Classical IP over ATM Logical IP Subnetworks (LIS) are created inside which all hosts communicate using the ATM network. Hosts from outside the LIS on other sub-nets can be reached only using IP-layer routing. All the IP traffic inside the LIS is mapped as best-effort (ATM UBR class). ATMARP (ATM Address Resolution Protocol) is used by edge devices to resolve IP addresses to native ATM addresses. For any pair of IP/ATM edge devices (hosts or routers) a single VC is created on demand and shared for all traffic between the two devices in *best effort* style. A new technology Multiprotocol Over ATM (MPOA), which offers QoS guarantees has been standardized under ATM Forum recently.

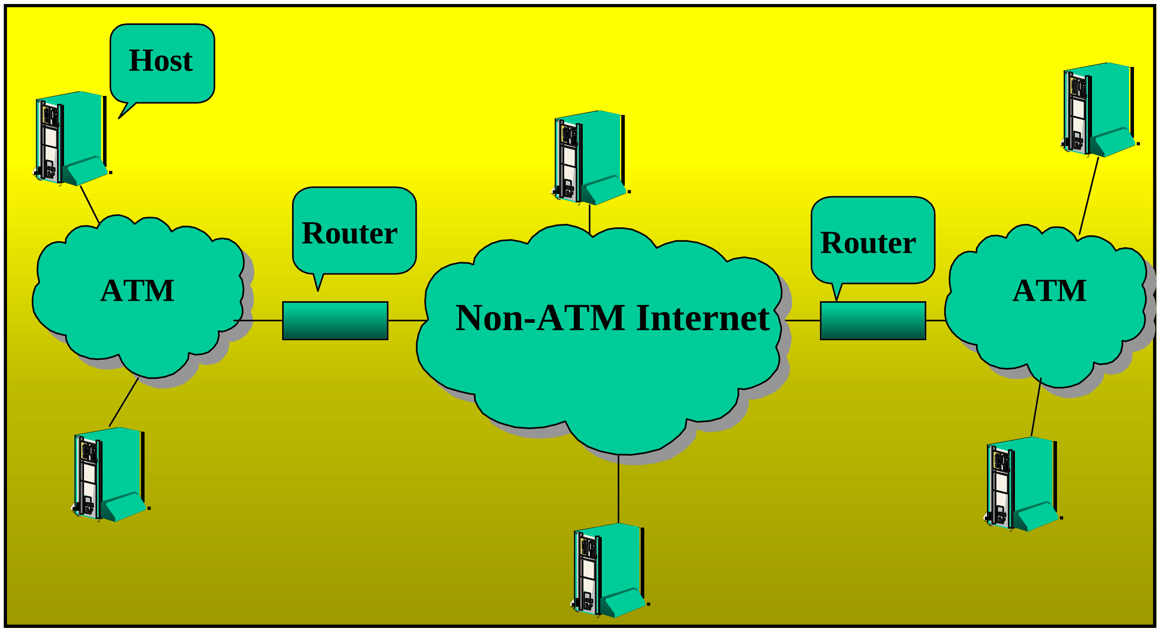


Figure 23: The IIS network architecture

The network architecture considered here is illustrated in Figure 23. The IP-attached hosts may send unicast traffic to other hosts or multicast traffic to all hosts that have joined a multicast group. The destination hosts may have used RSVP to reserve resources along the Internet path of the data flow. An ATM network lies along this path providing resources and QoS within the ATM cloud using VCs. The key device in the IIS over ATM integration is the egress router. It acts as both an IP router to the IP side of the network and as an ATM interface. It sets up, adds and tears down VCs. The edge device needs to have IP Integrated Services/RSVP capability, ATM UNI protocol capability and the capability to translate between the two. An IP-level reservation (RESV message) triggers the edge device to translate the RSVP service requirements into ATM VC semantics.

2.3.6.1. The Problem

There are a couple of problems that need to be solved in order to integrate IIS over ATM: IP multicast over ATM, integration of RSVP and ATM signaling. The first problem is soon solved with the Multicast Address Resolution Server, MARS. MARS compliments ATMARP by allowing an IP address to resolve into a list of native ATM addresses. The second problem of integrating the signaling of RSVP and ATM can be cut into two separate problems: QoS translation and VC management. QoS translation concerns mapping a QoS from the IIS model to a proper ATM QoS class. The VC problem involves decisions on how many VCs are needed and which traffic flows are routed over which VC [19].

2.3.6.2. IS to ATM QoS Class Translation

Issuing an ATM setup message includes the following information:

- Service category/Broadband Bearer capability
- AAL parameters
- Broadband Low Layer Information
- Calling and Called Party Addressing Information
- Traffic Descriptors
- QoS Parameters
- Additional TM/UNI 4.0 parameters

The TM/UNI 4.0 service categories are:

- Constant Bit Rate (CBR)
- Real-time Variable Bit Rate (rtVBR)
- Non-real-time Variable Bit Rate (nrtVBR)
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)

The most natural mapping for guaranteed service are CBR and rtVBR. They come from the fact that *guaranteed service (GS)* is a real-time service and needs timing support. The other ATM service categories can not provide delay estimates, nor can they guarantee consistently low delay for every packet. Of the two CBR and rtVBR rtVBR provides generally better use of the network resources and handles variable rate traffic better. The GS traffic descriptors are: peak rate p, a source Tspec rate r_s, a receiver Tspec rate r_r, and an Rspec rate R. The two Tspec rates are intended for supporting different receiver rates. At the moment this feature is still under study, and are assumed identical. The Tspec rate describes the traffic itself and is used for policing. The Rspec rate is the allocated service. If the receiver increases R over r it is in fact reducing delay. The bounds for traffic descriptor parameters for GS on rtVBR are:

$R \leq PCR \leq \min(p, \text{line rate})$

$r \leq SCR \leq PCR$

$0 \leq MBS < b$ (b = bucket depth parameter).

The *controlled load (CL)* service has a peak rate p, a Tspec rate r and a bucket depth parameter b. Appropriate service categories for CL are: CBR, nrtVBR and ABR. The traffic parameter bounds for nrtVBR are:

$$r \le SCR \le PCR \le \min(p, \text{line rate})$$

$$0 \le MBS < b.$$

For ABR VCs MCR parameter would be set according to Tspec. The bucket depth does not map onto an ATM parameter so the edge device must have a buffer of at least b bytes. For CBR the Tspec sets a lower bound for PCR and the edge device buffering must be adequately large to absorb all bursts.

The QoS parameters of TM/UNI 4.0 (Cell Loss Ratio CLR, Cell Transfer Delay CTD and Cell Delay Variation CDV do not have IP layer counterparts in IP services. Therefore they must be set by a policy in the edge device. ITU has defined a set of parameters for a number of QoS classes. Class 1 is appropriate for low-loss, low-delay CBR connections, and class 3 is appropriate for variable rate connections with loss and delay appropriate for non-real-time applications. Thus we can use QoS class 1 with GS, QoS class 3 with CL and QoS class 0, the unspecified QoS with best-effort traffic [20].

2.3.6.3. Integrated Services and ATM VCs with QoS

Here I will discuss the mapping of RSVP over ATM Switched Virtual Circuits, SVCs. This is based on an Internet draft by Berson, et. Al [19].

The first comments on RSVP over ATM were negative. The fundamental difference is that RSVP control is receiver oriented and ATM control is sender oriented. While this does impose some problems it is not a hopeless situation. The RESV requests of RSVP are generated at the receiver, but the actual reservations are allocated at the sub-net sender. For specific data flows this in fact means that sub-net senders will establish all QoS VCs and the sub-net receiver must be able to accept incoming QoS VCs. There are different approaches on how to actually accomplish this, but all of them that have VCs initiated and controlled by the sub-net senders will interoperate.

There are a number of different approaches on how to actually map different reservations on to VCs. The distinguishing factor is how the reservations are combined to individual VCs. Two opposite solutions are: 1)individual VCs for single reservation 2) reservations combined on VCs. The greatest benefit from the “non-aggregated” approach is ease of implementation. On the negative side we have increased VC setup time and consumption of a grater number of VC and associated resources. The aggregation model on the other hand is more difficult to implement, but it has lots of benefits. Traffic from multiple sources over multiple RSVP sessions could be multiplexed on to the same VC. There would be no signaling latency as the VCs were already setup before the traffic started flowing. Also the problem associated with multicast sessions: the support of different QoS for different users, referred as the “heterogeneity problem” would not be an issue. In the non-aggregated approach support for heterogeneous receivers in multicast could translate into setting up multiple VCs.

2.3.6.4. Implementation Requirements

The RSVP over ATM UNI 3.0 and 4.0 implementation requirements are according to [21]:

1. Heterogeneity support.
   - The implementation must not in the normal case send more than one copy of a particular packet to a particular ATM end-point. Implementations must also ensure that traffic is also sent to best-effort receivers.
2. Multicast data distribution.
   - A sender must set a service and not use global break bit(s) when using non-QoS supporting multicast servers.
3. Receiver transitions
   - When changing from one VC to another senders must send on the old VC or both the old and the new VC.
4. VC setup sender initiated
   - All RSVP triggered QoS VCs must be setup by the sub-net senders. All receivers must be able to accept incoming QoS VCs.
5. VC teardown controlled by RSVP
   - VC initiators must not tear down RSVP initiated VCs due to inactivity
   - VC receivers must not tear down any incoming VCs due to inactivity

The minimum requirements are:

1. Heterogeneity
2. Multicast end-point identification
3. Multicast data distribution
4. RSVP control VC management
5. Reservation to VC mapping
6. Dynamic QoS
   - Implementations must support RSVP initiated changes in reservation, so that existing VCs are replaced by new appropriately sized VCs.
7. Short-Cuts
   - Implementations should establish QoS short-cut whenever a best-effort short-cut is in use to a particular destination or next-hop. In other words best-effort short-cuts are never established. RSVP triggered short-cuts also should not be established

8. Encapsulation

   - Implementations must encapsulate data sent on QoS VCs with same encapsulation as is used on best-effort VCs.

3. Comparison of the Protocols

The goals for real-time communications are according to[22]:

- low delay variance
- low latency
- ability to easily integrate non real-time and real-time services
- adaptable to dynamically changing network and traffic conditions
- good performance for large networks and large number of connections
- modest buffer requirements within the network
- high effective bandwidth utilization
- low overhead in header bits per packet or cell
- low processing overhead per packet within the network and at the end system

Here we will compare the protocols presented in chapters one and two. The list of real-time communication goals will be kept in mind as well as the issues regarding delay variance and synchronization. I will also give some comments on the possible applicability and future of these protocols here and in the conclusions.

3.1.1. Transport Layer Protocols: TCP vs. RTP/UDP

The transport layer alternatives for real-time communication currently are TCP and RTP/UDP. TCP is a protocol that guarantees end-to-end deliveries with retransmission. As discussed earlier this is in general bad for real-time communication. TCP also adds overhead, in growing headers. TCP does not reduce delay variance in any way. On the other hand TCP does provide some adaptation to changing network condition through retransmission and it provides good scaling. A combination of RTP/TCP/IP is not all together bad. In audio and video on demand applications where the end host has the ability of buffer several seconds of data, using TCP can avoid all gaps and glitches in continuous data. But as was mentioned in 1.3. RTP/UDP/IP is a better choice when delay needs to be minimized and thus buffering before playback is not desired.

3.1.2. Session Layer Control Protocols: RTP vs. MSTP

There are many similarities in RTP (1.3) and MSTP the implementation (1.2) of RTSM (1.1.1). Both rely on session level in delay variance avoidance and both try to synchronize different media. MSTP goes further in being an application independent solution while RTP must always be implemented in the application. The synchronization model MSTP uses is also more novel than RTP’s. The key- and time-media concept smoothes glitches in the case where application is ready to playback the key-media (audio), but is still waiting for video to arrive to maintain synchronization. On the negative side of MSTP is that it adds overhead. A video conferencing application could choose to use RTP and deploy the key-media concept by making audio the key- and time-media (which is actually the case in today's applications). Still it can not be denied that the performance measurements of MSTP in [1] are quite impressive.

3.1.3. Reservation Protocols: ST-II+ vs. IIS RSVP vs. Tenet 2

Many things can be said about the overall feasibility of resource reservation protocols in the Internet of today or tomorrow. They are fundamentally very complicated, the charging is a difficult task to implement and is still mostly an open question, they re-do many of the tasks that the underlying physical network of choice, ATM already does, they cause unnecessary overhead and they scale poorly (if they scale at all) to an environment where new connections are set up and torn down 10.000 times every second. Other very promising new concepts are coming out: [23, 24, 25] , but they are at the moment in a preliminary stage. Therefore I will concentrate on ST-II, the IIS (RSVP) and the Tenet Suite.

There is a fundamental difference in the way ST-II and RSVP reserve resources in routers. ST-II uses a simplistic point to point architecture that is very inefficient in terms of network resource allocation in the case of multicast communications as is shown in [26]. RSVP incorporates heterogeneous receiver requests and multiple reservations styles provide additional opportunities to improve network-wide resource utilization.

RSVP reservation of resources is receiver oriented. In a multicast session the receivers can reserve resources based on its individual needs and capabilities. ST-II+ and Tenet are source oriented. The source reserves resources for its traffic flow. The receiver based method has been proven to scale better for large multicast sessions, and it also simplifies charging for reserved resources in a broadcast type session. Dynamic addition of receivers under ST-II requires the generation of Connect and accept messages between source and receiver. This results in overhead that is proportional to the number of downstream receivers. What RSVP does is that it merges new request to existing distribution branches. The join overhead of ST-II becomes significant already with ten receivers.

ST-II provides multicast routing functions itself by building a multicast distribution tree based upon unicast routing tables and performs the replication and forwarding of packets itself. RSVP assumes that the underlying network provides the multicast functions. The functions ST-II does in order to do multicast adds some processing overhead, but does not affect resource reservations or protocol messaging overhead.

The soft-state vs. hard-state approaches of ST-II, Tenet 2 and RSVP can be argued in how they provide reliability and robustness in the face of changing network conditions, but are in fact difficult to compare. ST-II hello interval and RSVP refresh messages both use timers, and the timer values are not specified in the protocol standards. However, the design philosophies can be compared. ST-II failure detection that uses messages Hello, Status and Notify adds complexity to the protocol. RSVP soft-state approach is very "Internet like" in being light and leaving all responsibility to the end hosts. ST-II requires the networks to be responsible for correctness. On the other hand RSVP causes refresh overhead and occasional quality of service disruptions can occur due to sudden routing changes.

IP/RSVP separates data transfer and the resource reservation into two protocols. The Tenet approach is similar - RCAP is used for channel establishment and tear down and the RTIP/RMTP combination is used for data transport. ST-II+ combines the two functions into one protocol. There is no clear architectural preference here, however the separation of control and delivery protocols seem to facilitate development [11].

The traffic classes in the real-time suites are slightly different. Tenet provides deterministic guarantees with delay and delay variance bounds for those applications that can be classified as hard-real-time application and do not tolerate occasional packet loss. Packets can be lost because of buffer overflow at the destination or in the network, or from late arrivals at the destination. The IS architecture guaranteed service differs from the Tenet in that it does not provide bounds on delay variance. The predictive service of Tenet and the IS architecture is similar: a service that provides statistical bounds on delay and delay-loss. What the actual service will be like depends on the admission control and scheduling implementation (see 2.3.4). The IS architecture leaves the actual implementation open. It is a service provider policy issue.

The current implementation of the Tenet suite uses number of different rate based service disciplines: 1) scheduler based: Delay Earliest Due Date (EDD-D), Delay variance Earliest Due Date (EDD-J) 2)rate based: rate-controlled static priority (RCSP). The ISA implementation proposal by [23] uses WFQ with FIFO+ inside traffic classes with same priority. The Tenet Suite is modular and can accommodate additional scheduling disciplines. The EDD-J and RCSP methods minimize buffer space and delay variance. With EDD-J quarantees on delay and delay variance can be provided as long as the schedulability criterion is met (link utilization less than 100% with EDD). RCSP and WFQ give tight guarantees on delay and delay variance bounds and the implementation is also more straightforward than that of the scheduler based.

Conclusions

A number of protocols that address the needs of real-time multimedia communications were introduced in this paper. RTP and MSTP are protocols that give session level synchronization. However, MSTP is a protocol for research purposes, and as RTP is already widely deployed it seems to have only academic value. Also a couple of real-time protocol suites were presented: ST-II+, Tenet Suite 2 and the IETF Internet Integrated Services Architecture and its components concept were presented. The IIS seems to be hot right now with a lot of academic and industry research effort going into it. All of the three schemes share at least on thing in common: they are gigantically complex and have no understandable charging schemes designed yet.

In chapter 3 the multimedia protocols are compared against the design goals. There are no clear winners - all protocols have pros and cons. The design of a protocol involves lots of parameters and many compromises between them have to be made. Examples of such are those between the header overhead and extensibility and flexibility.

The future of the multimedia protocols is unclear. RTP seems to be the only definite winner at the moment. If the new technologies like ATM Forums MPOA, Ipsilon’s IP Switching and Cisco’s Tag Switching as such provide adequate for the real-time applications something as complicated as RSVP might not be needed.

References

[1]Yang, C-C., Huang J-H., "A Multimedia Synchronization Model and Its Implementation in Transport Protocols", IEEE Journal on Selected Areas is Communications, vol.14, no.1, pp. 212- 225, Jan. 1996.

[2] Anderson ,D.P., Homsy, G. “A Continuous Media Transport and Orchestration service”, ACM SIGCOMM,pp.99-110, 1992.

[3] Schulzrinne, H., “Issues in Designing a Transport protocol for Audio and Video Conferences and other Multiparticipant Real-Time Applications”, Audio-Video Transport Working Group Internet-draft , Internet Engineering Taskforce, October 1993.

[4] Shulzrinne, H., Casner., S., Frederick, R., Jacobsen, V., “A Transport Protocol for Real-Time Applications”, Network Working Group Request for comments:1889, Internet Engineering Task Force , Jan. 1996.

[5] ITU-T, “H.225.0 Line Transmission of Non-Telephone Signals”,pp.9-118, May 1996.

[6] Bolot, J.-C., Vega-García, A., “The Case for FEC-Based Error Control for Packet Audio in the Internet”, to be published in ACM Multimedia Systems.

[7] Bolot, J.-C., Vega-García, A., “Control Mechanisms for Packet Audio in the Internet, Proc. IEEE Infocom ’96, San Francisco, California, pp.232-239, Apr. 1996.

[8] Bolot, J.-C., Hugues, C., “Analysis and Control of Audio Packet Loss over Packet-Switched Networks”, Inria, Sophia Antinopolis, France, 1993.

[9] Gupta, A., "Multi-party Real.Time Communications in Integrated Services Packet-Switching Networks", The Tenet Group, Computer Science Division, Berkeley, 1996, http://http.cs.berkeley.edu/mit/rescearch/qual-proposal/.

[10] Delgrossi, L., Berger, L., “Internet Stream Protocol Version 2 (ST-2), Protocol Specification - Version ST-2+”, Network Working Group Request For Comments: 1819, Internet Engineering Taskforce, Aug. 1995.

[11] Banerjea, A., Ferrari, D., Mah, B., Moran, M., Verma, D., and Zhang, H., "The Tenet Real-Time Protocol Suite: Design, Implementation And Experiences", IEEE/ACM Transactions on Networking, vol. 4, no.1, pp.1-10, Feb 1996.

[12] Shenker, S., Partridge, C., Guerin, R., “Specification of Guaranteed Quality of Service”, Integrated Services Working Group, Internet-Draft, Feb.1997, draft-ietf-intserv-guaranteed-svc-07.txt.

[13] Wroclawski, J., “Specification of the Controlled-Load Network Element Service”, Integrated Services Working Group, Internet-Draft, Nov..1997, draft-ietf-intserv-ctrl-load-svc-04.txt.

[14] Crowcoft, J., Wang, Z., Smith, A., Adams, J., “A Rough Comparison of the IETF and ATM Service Models”, IEEE Network, Vol.9, No.6, Nov./Dec. 1995, pp.12-16

[15] Braden, R., Clark, D., Shenker, S., “Integrated Services in the Internet Architecture: An Overview”, Network Working Group, Request for Comments:1633, Jun. 1994.

[16] Wroclawski, J., “The Use of RSVP with IETF Integrated Services”, Work in Progress, Integrated Services Working Group, Oct. 1996, draft-ietf-inserv-rsvp-use-01.txt.

[17] Braden, R., et. Al, “Resource ReSerVation Protocol (RSVP) – version 1 Functional Specification”, Internet draft, Nov. 1996, draft-ietf-rsvp-spec-14.txt.

[18] Rajagoplan, B., Nair, R., “Quality of Service-Based Routing in the Internet - Some Issues”, Internet draft, Oct. 1996, draft-nair-qos-based-routing-01.txt.

[19]Berson, S., Berger, L., “IP Integrated Services with RSVP over ATM”, Internet Draft, Mar. 1997, draft-ietf-issll-atm-support-03.txt.

[20] Garrett, M., Borden, M., ”Interoperation of Controlled-Load and Guaranteed-Service with ATM”, Internet Draft, Mar. 1997, draft-ietf-issll-atm-mapping-02.txt.

[21] Berger, L., “RSVP over ATM Implementation Guidelines”, Internet Draft, Mar. 1997, draft-ietf-issll-atm-imp-guide-00.txt.

[22] Aras, C., Kurose, J., Reeves, D., Shultzrinne, H., “Real-Time Communication in Packet-Switched Networks”, Proceedings of IEEE, vol.82, no.1, pp. 122- 138, Jan.1994

[23] Clark, D., “Adding Service Discrimination to the Internet”, MIT, Laboratory of Computer Science, Sept. 1996, http://ana-www.lcs.mit.edu/anaweb/pdf-papers/TPRC2-0.pdf

[24] Kilkki, K., “Simple Integrated Media Access”, Internet Draft, Internet Engineering Task Force, Networking Group, Mar. 1997, draft-kalevi-simple-media-access-00.txt.

[25] Luoma, M., Ilvesmäki. M., “Simplified Management of ATM traffic”, Proc. SPIE’97, Voice, Video and Data Communications, Dallas, TX, Nov. 1997.

[26] Mitzel, D., Estrin, D., Shenker, S., Zhang, L., “An Architectural Comparison of ST-II and RSVP”, University of Southern California, Informations Sciences Institute, 1996.